This application is submitted in the name of Inventor Colin Whitby-Strevens, assignor to Apple Computer, Inc. a California Corporation.

SPECIFICATION

CYCLEMASTER SYNCHRONIZATION IN A DISTRIBUTED BRIDGE

FIELD OF THE INVENTION

[0001] The present invention relates broadly to synchronizing devices in communication with each other over a serial bus connection. Specifically, the present invention relates to synchronizing cyclemasters across a distributed bridge architecture in a IEEE 1394-compliant network.

BACKGROUND OF THE INVENTION

[0002] The Institute of Electrical and Electronic Engineers (IEEE) has promulgated a number of versions of a high speed serial bus protocol falling under the IEEE 1394 standard (referred to herein collectively as "1394"). A typical serial bus having a1394 architecture interconnects multiple node devices via point-to-point links, such as cables, each connecting a single node on the serial bus to another node on the serial bus. Data packets are propagated throughout the serial bus using a number of point-to-point transactions, such that a node that receives a packet from another node via a first point-to-point link retransmits the received packet via other point-to-point links. A tree network configuration and associated packet handling protocol ensures that each node receives every packet once. The 1394-compliant serial bus may be used as an alternate bus for the parallel backplane of a computer system, as a low cost peripheral bus, or as a bus bridge between architecturally compatible buses.

[0003] The 1394 standard specifies two primary types of bus access: asynchronous access and isochronous access. Asynchronous access may be described as either "fair" or "priority." Priority access is used by nodes that need the next available asynchronous opportunity to transfer data. Isochronous access is used by nodes that require guaranteed bandwidth with bounded latency, for example, nodes transmitting video or audio data. The transactions for each type of bus access are comprised of at least one subaction, wherein a subaction is a complete one-way transfer operation.

[0004] In the case of digital video data transfers within 1394-compliant systems, the video data may be transferred between a mass storage device and a digital video camera or other recorder under the control of a computer processor or other device. The video data is transferred as a series of frames, each frame being made up of a number of data packets. The individual data packets include a number header fields as well as the video data itself. In order to ensure that each frame of the video data is played out in the proper sequence, the frames are time stamped with an appropriate frame presentation time measured in terms of cycle time of an isochronous transaction on a 1394-compliant bus when they are recorded. The cycle time is maintained by a cyclemaster as described in the 1394 standard. The cyclemaster uses priority access to broadcast a cycle start packet. This initiates an isochronous cycle, during which nodes can use isochronous access, and contains the cyclemaster's cycle time clock information so that all nodes associated maintain the necessary synchronization for audio and video data.

[0005] Bus bridges between multiple buses of devices forward request and response subactions from one bus to another, allowing transaction requester and responder components to be located on different buses. Each bus has its own cyclemaster. An exemplary 1394-compliant network of three buses of devices is illustrated in FIG. 1. The first bus includes devices a1, a2, and a3 coupled together by 1394-compliant cables. Specifically, device a1 is coupled to device a2 by cable 40. Device a2 is coupled to device a3 by cable 42. Device a3 is then coupled to portal 22 of bridge 20. Bridge 20 couples the first bus to a second bus, which includes the devices b1, b2 and b3. Portal 24 of bridge 20 is coupled to device b1 by cable 46. Device b1 is coupled to device b2 by cable 48. Device b1 is coupled to device b2 by cable 48. Device b2 is coupled to device b3 by cable 50. Device b3 is coupled to portal 32 of bridge 30. Bridge 30 couples the second bus to a third bus that includes the devices c1 and c2. A second portal 34 of bridge 30 is coupled to device c1 by cable 54. Device c1 is coupled to device c2 by cable 56. Bridges 20 and 30 allow devices on the different buses to communicate with each other using both asynchronous and isochronous communications. For example, if device a1 sends a communication to device b3, the communication is passed along the first bus until it reaches portal 22 of bridge 20. Bridge 20 then recognizes that the communication is addressed to the device b3 and forwards the communication from portal 22 to portal 24.

[0006] The 1394 standard requires that 1394 bridges implement a method by which all the cyclemasters in a network are kept in phase. The topology used to model the method is shown in FIG. 2. One cycle master is elected to be the master or net cyclemaster. Cyclemasters on buses directly attached to the bus with the net cycle master are kept in phase with the net cycle master. In turn, other buses attached to these buses are kept in phase with the cycle masters on these buses, and so on. Each bridge is responsible for calculating the amount by which the cyclemaster on the bus on its alpha portal (the portal away from the net cyclemaster is out of phase with the cyclemaster on the bus on its other portal, and accordingly issuing cyclemaster adjustment commands to the cyclemaster on its alpha portal to shorten or lengthen the cycle by one *cycle_offset* unit (40 ns). According to the method of the 1394 standard, the phase difference is sampled once during each isochronous period. The bridge simultaneously samples the value of CYCLE_TIME.*cycle_offset* for both of its portals and subtracts, modulus 3072, the upstream portal's cycle offset from the alpha portal's cycle offset.

[0007] FIG. 3 illustrates how the delay from the cyclemasters to the respective portals is considered by the 1394 standard method. The two cycle timers connected to the bridge provide, when simultaneously sampled, CYCLE_TIME.*cycle_offset* values denoted as upstreamOffset and alphaOffset respectively. In the example illustrated in FIG. 3, the upstream delay is 42 units, and the alpha delay is 17 units (as per the example in section 4.4 of the IEEE Serial Bus Protocol 1394.1). When the two cycle timers are simultaneously sampled, the correction to be applied to the alpha cyclemaster is (208+17) – (193+42) = -10. This negative value is interpreted as a "go faster" command, which causes the alpha cyclemaster to wrap at 3070 instead of 3071. The expectation is that on the next isochronous cycle, the difference will then be –9, and the method eventually brings the difference to a zero value.

[0008] However, in a distributed bridge, where the two portals are connected by a long haul or wireless medium, there may be no common clock to be sampled by the cycle timers simultaneously, so the method taught by the 1394 standard is useless. It is now desirable to attach devices by wireless connections to 1394 buses, as well as by significantly longer cable lengths. Thus, there is a heartfelt need to provide synchronization of cyclemasters that facilitate connecting wireless or longhaul connections.

SUMMARY

[0009] The present invention addresses the problem of wireless or longhaul connections and provides cyclemaster synchronization over a distributed bridge. In a distributed bridge, when a cycle synchronization event occurs on a portal, such as when the *cycle_offset* rolls over, the portal sends a synchronization signal to its peer portal through the bridge fabric. When the peer portal receives such a synchronization signal it samples the local cycle timer. The bridge management software at the upstream portal communicates the sample value to the bridge management software at the alpha portal. The bridge management software at the alpha portal is then able to use the sampled time values to compensate for the delays through the bridge fabric, and calculate the correction to be applied to the alpha cycle timer. Embodiments of the present invention assume that the delay through the bridge fabric is constant, and the same in both directions.

[0010] In one aspect, the present invention provides a method of synchronizing cyclemasters over a distributed bridge. A local portal sends a synchronization signal to a peer portal through a bridge fabric upon occurrence of a cycle synchronization event on the local portal. The peer portal samples its local cycle timer to obtain a sample value when the peer portal receives the synchronization signal. A bridge manager at an upstream portal communicates the sample value to a bridge manager at an alpha portal. The bridge manager at the alpha portal uses the sampled time value to compensate for delays through a bridge fabric, calculate the correction to be applied to a cycle timer associated with the alpha portal, and correct the cycle timer. In an embodiment, the cycle synchronization event comprises a cycle offset value rolling over.

[0011] In another aspect, the present invention provides alternative method of synchronizing cyclemasters over a distributed bridge. An output signal means from a first portal is connected with an input signal means of a second portal and an output signal means from a second portal is connected with an input signal means of a first portal. The output signal means of the first portal is sampled and stored. The sampled value is communicated to a downstream portal. The downstream portal adjusts its cyclemaster in response to the sampled value. In an embodiment, an interrupt is generated when the output signal means is sampled.

[0012] In yet another embodiment, the present invention provides a method of synchronizing cyclemasters over a distributed bridge, by performing the acts of: connecting an output signal means from a first portal with an input signal means of a second portal and connecting an output signal means from a second portal with an input signal means of a first portal; sampling the output signal means of the first portal and storing the sampled value; communicating the sampled value to a downstream portal; if the sampled value is received by the downstream portal within an acceptable time period, the downstream portal adjusting its cyclemaster in response to the sampled value; and if the sampled value is not received within an acceptable time period, then communicating an error condition that indicates that cyclemaster adjustment cannot be performed. In an embodiment, cyclemaster adjustment is abandoned for a current isochronous cycle if the sampled value is not received within an acceptable time period. In an embodiment, the error condition is indicated by a bridge fabric connecting the first and second portals, and can be indicated by setting error flags.

[0013] In still another embodiment, the present invention provides a bridge link device, connectable within a 1394-compliant serial bus architecture. The bridge link device comprises a first sampled value reflecting an output signal value; and a second sampled value reflecting an input signal value; a sample value register, the sample value register containing the first sampled value and the second sampled value, the sample value register in communication with software that communicates the sampled values to a downstream node device.

[0014] Many other features and advantages of the present invention will be realized upon reading the following detailed description, when considered in conjunction with the accompanying drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] FIG. 1 illustrates in block diagram form a 1394-compliant network of buses connected by bridges;

[0016] FIG. 2 illustrates a bridge connecting buses having separate cyclemasters;

[0017] FIG. 3 is a timing diagram that illustrates how cyclemaster synchronization is performed in accordance with prior art 1394 methods; and

[0018] FIG. 4 is a timing diagram that illustrates how cyclemaster synchronization in a distributed bridge is performed in accordance with the present invention.

[0019] FIG. 5 is a flowchart illustrating a sequence of acts executed in accordance with an embodiment of the present invention.

[0020] FIG. 6 is a flowchart illustrating a sequence of acts executed in accordance with another embodiment of the present invention.

DETAILED DESCRIPTION

[0021] Embodiments of the present invention utilize the following calculations. al_co represents the alpha portal cycle offset at any moment in time. up_co represents the upstream portal cycle offset at any moment in time. d represents the real difference between the two cycle offset values (al_co – up_co). For the time period covered by the sampling time of the algorithm, d is assumed constant, and represents the difference between the "simultaneous sample" values as expressed in the 1394.1 algorithm. fd represents the fabric delay (assumed constant and the same in both directions). up_co' represents the sampled value of the upstream portal cycle offset when the sampling signal is received from the alpha portal. Then,

$$up\_co' = d + fd \qquad (1)$$

al_co' represents the sampled value of the alpha portal cycle offset when the sampling signal is received from the upstream portal. Then, al_co = -d + fd (2)

Subtracting (1) from (2) eliminates the fabric delay

al_co- up_co' = -d + fd – (d + fd) = -2d (3)

Hence d = -(al_co – up_co')/2 (4)

[0022] FIG. 4 illustrates the application of formulae (1)-(4), where d = -15, and fd = 38. In this embodiment, the upstream portal samples the value 23 when it receives the sampling signal from the alpha portal, and communicates this value to the alpha portal. The alpha portal samples the value 53 when it receives the sampling signal from the upstream portal. When it receives the value 23 from the upstream portal, the difference is calculated as -(23-53)/2 =15. In a preferred embodiment, the arithmetic is performed modulo 3072. The isochronous cycle repeats on a nominal 8Kz clock (125 usec). The cyclemaster clock is itself a 25Mhz clock. 3072 ticks of this clock measures 125 microseconds. Thus in normal cyclemaster behavior when the cycle offset reaches 3072, a new isochronous cycle is started and the cycle offset reset to zero.

[0023] In an embodiment of the present invention, bridge link hardware incorporates an output sample signal, an input sample signal, and a software readable register for storing the sampled values. Directing attention to FIG. 5, the bridge link fabric connects the four signals (output from one portal to the input of the other portal and vice versa) with a constant delay (act 500). Jitter on the delay is bounded by 20ns-40ns, preferably closer to 20ns. The value of the delay is immaterial, except that the sampled value is subsequently accessed by software in the upstream portal (act 502) and communicated to the downstream portal (act 504) in sufficient time for the downstream portal to issue the cycle adjustment command to the cyclemaster (act 504). In the preferred embodiment, the bridge portal hardware also generates an interrupt signal when it takes a sample, and so is ready for software to read the sample and either send it to the alpha portal (upstream portal) or perform the cycle adjustment calculation (alpha portal). An indication, such as error flag, is made by the bridge fabric if either of the sampled signals is not delivered on time. Software uses the indication to abandon cyclemaster adjustment for the current isochronous cycle. Leaving the cyclemaster unadjusted is not problematic if error indications occur rarely. Alternatively, a double-sample scheme similar to the one described below may be used to provide redundancy.

[0024] In another embodiment of the present invention, a combined hardware and software implementation can be utilized to provide synchronization of cyclemasters over a distributed bridge. Directing attention to FIG. 6, in this embodiment, the portals have two sample inputs and two registers. The bridge fabric sends sampling signals (act 600) to the local bridge portal at the moment that a message is launched across the bridge fabric and at the moment that a message arrives from the bridge fabric. These signals are used to trigger the sampling mechanisms in the bridge. The alpha portal generates a cycle synchronization interrupt to the local bridge management software (act 602). This queues an appropriate command message to send to its peer across the bridge fabric (act 604). At the moment that the message is launched (act 606), the bridge fabric hardware triggers a *cycle_offset* sample in the alpha portal bridge hardware (act 608). At the moment that the message arrives at the upstream portal, the bridge fabric hardware triggers the *cycle_offset* sample in the upstream portal (act 610). The alpha portal uses the local sample to compensate for the software delay in sending the sampling message (act 612). Similarly, the upstream portal takes a local sample at the moment that it sends its sampling message to the alpha portal (act 614). Software in the upstream portal subsequently communicates this value to the software in the alpha portal (act 616), so that software in the alpha portal can compensate for the software delay in the upstream portal in sending the sampling message to the alpha portal (act 618). At this point, the alpha portal now has enough information to determine whether an adjustment to the cyclemaster on its local bus is necessary, and if so command that adjustment. The signals used by the bridge fabric hardware can be readily derived from existing signals. For example, the sampling signal on message arrival can be the same signal as is used to generate an interrupt. Care needs to be taken that the samples are not overwritten by subsequent message transmissions/arrivals before they have been read by software. However, taking samples, for example, on every message, regardless of whether the message is a sampling message or not, is relatively safe. If a message transmission fails for some reason, then it can be retried. In a preferred embodiment, a polling scheme performs the retries and sets a flag when message transmission is complete. If the bridge fabric supports automatic retries, which may affect delivery latency, then the sampling signal for transmit is reasserted.

[0025] While embodiments of methods and apparatus for synchronizing cyclemasters over distributed bridges has been illustrated and described, it is to be understood that many variations can be made to embodiments of the present invention, without departing from the spirit thereof.